UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,733	02/27/2004	John W. Curry	200314830-1	8425

22879 7590 11/20/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
MEHRMANESH, ELMIRA	

ART UNIT	PAPER NUMBER
2113	

NOTIFICATION DATE	DELIVERY MODE
11/20/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	Application No.	Applicant(s)	
	10/789,733	CURRY, JOHN W.	
	Examiner	Art Unit	
	Elmira Mehrmanesh	2113	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 July 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This action is in response to an amendment filed on July 28, 2008 for the application of Curry, for a "Detecting floating point hardware failures" filed February 27, 2004.

Claims 1-27 are pending in the application.

Claims 1-27 are rejected under 35 USC § 103.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erskine et al. (U.S. Patent No. 6,564,162) in view of Breslau et al. (U.S. Patent No. 6,101,501).

As per claim 1, Erskine discloses a method for testing floating point hardware in a processor while executing a computer program (Fig. 1), comprising:

executing a first set of code of said computer program (col. 3, lines 60-62), said first set of code having a first floating point instruction (col. 3, lines 45-49), thereby obtaining an emulated result (col. 3, lines 60-62, *software emulation*)

executing said first floating point instruction utilizing said floating-point hardware, thereby obtaining a hardware-generated result (col. 4, lines 54-55 and Fig. 1, element 50)

comparing said emulated result with said hardware-generated result (col. 4, lines 65-67 and Fig. 1, element 70).

Erskine discloses software emulation (col. 3, lines 60-62); however he fails to explicitly disclose not using floating point hardware.

Breslau teaches:

executing a first set of code of said computer program without employing said floating point hardware (col. 8, lines 51-67).

It would have been obvious to one of ordinary skill in the art at the time the invention to use the method emulation of floating point operations of Erskine et al.'s in combination with the software emulation of Breslau et al.

One of ordinary skill in the art at the time of the invention would have been motivated to make the combination because both inventions disclose a method and system for emulating software/hardware in a processing device (Erskine, col. 3, lines

60-62 and col. 4, lines 54-55) and (Breslau, col. 78, lines 35-45). Erskine discloses enhancing productive debugging by modifying specific elements (i.e. register values) (col. 14, lines 22-29) and he further discloses various changes could be applied to his invention (col. 14, lines 42-46). Breslau's system provides further details as to switch to software emulation or use of hardware (col. 8, lines 51-67).

As per claim 23, Erskine discloses an article of manufacture comprising a program storage medium having computer readable code embodied therein (col. 3, lines 31-36), said computer readable code being configured to test floating point hardware in a processor while executing a computer program (Fig. 1), comprising:

computer readable code (col. 3, lines 31-36) for executing a first set of code of said computer program (col. 3, lines 60-62), said first set of code having a first floating point instruction (col. 3, lines 45-49), thereby obtaining an emulated result (col. 3, lines 60-62, *software emulation*)

computer readable code (col. 3, lines 31-36) for executing said first floating point instruction utilizing said floating-point hardware, thereby obtaining a hardware-generated result (col. 4, lines 54-55 and Fig. 1, element 50)

computer readable code (col. 3, lines 31-36) for comparing said emulated result with said hardware-generated result (col. 4, lines 65-67 and Fig. 1, element 70).

Breslau teaches:

executing a first set of code of said computer program without employing said floating point hardware (col. 8, lines 51-67).

Claims 2, 4-7, 9-14, 16-19, 21, 22, 24-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erskine et al. (U.S. Patent No. 6,564,162) in view of Breslau et al. (U.S. Patent No. 6,101,501) and in further view of Van Dyke et al. (U.S. Patent No. 7,047,394).

As per claim 2, Van Dyke discloses rendering said floating point hardware unavailable prior to said executing said first set of code of said computer program without employing said floating point hardware (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

It would have been obvious to one of ordinary skill in the art at the time the invention to use the method emulation of floating point operations of Erskine et al.'s in combination with the floating-point computation processor of Van Dyke et al.

One of ordinary skill in the art at the time the invention would have been motivated to make the combination because both inventions disclose a method and system for emulating software/hardware in a processing device (Erskine, col. 3, lines 60-62 and col. 4, lines 54-55) and (Van Dyke, col. 73, lines 42-47) by changing register values (Erskine, col. 6, lines 1-18) and (Van Dyke, col. 32, lines 58-63) through utilizing the PA-RISC computer architecture, which provides highly regular instruction formats (Erskine, col. 18, lines 21-27) and (Van Dyke, col. 22, lines 51-59). Erskine discloses enhancing productive debugging by modifying specific elements (i.e. register values) (col. 14, lines 22-29) and he further discloses various changes could be applied to his invention (col. 14, lines 42-46). Van Dyke's system provides further details as to

changing the register values to enable/disable the emulation/debugging (col. 32, lines 58-63 and col. 73, lines 42-49).

As per claim 4, Van Dyke discloses processor represents a PA-RISC.™ processor (Fig. 1A, element 120) said rendering said floating-point hardware available including setting a CR10 co-processor control register of said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 5, Van Dyke discloses rendering said floating point hardware unavailable includes writing a first predefined value into a register in said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 6, Van Dyke discloses rendering said floating point hardware available for executing instructions of said computer program prior to said executing said first floating point instruction utilizing said floating point hardware (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 7, Van Dyke discloses rendering said hardware available includes writing a second predefined value into said register in said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 9, Van Dyke discloses processor represents a PA-RISC.™ processor (Fig. 1A, element 120) said rendering said floating-point hardware available including setting a CR10 co-processor control register of said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 10, Van Dyke discloses obtaining said hardware-generated result includes obtaining a trap result after said first floating point instruction is executed utilizing said floating point hardware (col. 39, lines 5-15).

As per claim 11, Van Dyke discloses obtaining emulated result includes obtaining a hardware-generated trap result after said processor encounters said first floating point instruction while said floating point hardware is unavailable (col. 38, lines 61-67 through col. 39, lines 1-4).

As per claim 12, Van Dyke discloses computer program represents a field application program (col. 110, lines 42-44).

As per claim 13, Erskine discloses a method for detecting failure in floating point hardware of a processor while executing a computer program (Fig. 1), comprising:

executing a first floating point operation of said computer program by emulating said floating point operation (col. 3, lines 60-62), thereby obtaining an emulated result (col. 3, lines 60-62, *software emulation*)

executing said first floating point operation utilizing said floating-point hardware, thereby obtaining a hardware-generated result (col. 4, lines 54-55 and Fig. 1, element 50)

and comparing said emulated result with said hardware-generated result to detect said failure to detect said failure (col. 4, lines 65-67 and Fig. 1, element 70)

Erskine discloses software emulation (col. 3, lines 60-62), however he fails to explicitly disclose not using a set of non-floating point operations.

Breslau teaches:

executing a first floating point operation of said computer program by emulating said floating point operation with a set of non-floating point operations (col. 8, lines 51-67).

Erskine discloses a debugging process (Fig. 1 and col. 14, lines 22-29); however he fails to explicitly disclose entering a diagnostic mode.

Van Dyke teaches:

entering a diagnostic mode (col. 73, lines 42-49, enable/disable, debugging)

determining whether diagnostic mode is to be continued and resuming execution of said computer program in a non-diagnostic mode (col. 118, lines 51-56)

if said diagnostic mode is to be discontinued, said non-diagnostic mode involving performing floating point operations of said computer program without emulating with non-floating point operations (col. 89, lines 66-67 through col. 90, lines 1-5 and col. 110, lines 42-44).

As per claim 14, Van Dyke discloses rendering said floating point hardware unavailable prior to said executing said first floating point operation by said emulating (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 16, Van Dyke discloses processor represents a PA-RISC.™ processor (Fig. 1A, element 120) said rendering said floating-point hardware available including setting a CR10 co-processor control register of said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 17, Van Dyke discloses rendering said floating point hardware unavailable includes writing a first predefined value into a register in said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 18, Van Dyke discloses rendering said floating point hardware available for executing instructions of said computer program prior to said executing said first floating point operation utilizing said floating point hardware (col. 73, lines 42-49 and col. 84, lines 23-27).

As per claim 19, Van Dyke discloses hardware available includes writing a second predefined value into said register in said processor (col. 73, lines 42-49 and col. 84, lines 23-27).

As per claim 21, Van Dyke discloses processor represents a PA-RISC.™ processor (Fig. 1A, element 120) said rendering said floating-point hardware available including setting a CR10 co-processor control register of said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 22, Van Dyke discloses hardware-generated result includes obtaining a trap after said first floating point operation is executed utilizing said floating point hardware (col. 39, lines 5-15).

As per claim 24, Van Dyke discloses computer readable code for rendering said floating point hardware unavailable prior to said executing said first set of code of said computer program without employing said floating point hardware (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 25, Van Dyke discloses computer readable code for rendering said floating point hardware unavailable includes computer readable code for writing a first predefined value into a register in said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 26, Van Dyke discloses computer readable code for rendering said floating point hardware available for executing instructions of said computer program prior to said executing said first floating point instruction utilizing said floating point

hardware (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

As per claim 27, Van Dyke discloses computer readable code for rendering said hardware available includes computer readable code for writing a second predefined value into said register in said processor (col. 32, lines 58-63 and col. 73, lines 42-49, *enable/disable*).

Claims 3 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erskine et al. (U.S. Patent No. 6,564,162) in view of Breslau et al. (U.S. Patent No. 6,101,501) and in further view of Markstein et al. (U.S. PGPUB 20040158600).

As per claims 3 and 8, Erskine in view of Breslau et al. fails to explicitly disclose an Itanium™ processor.

Markstein teaches:

processor represents an Itanium™ processor, said rendering said floating point hardware unavailable including setting at least one of a DFH and a DFL bit in a processor status register of said processor (page 3, paragraph [0035] and [0036]).

It would have been obvious to one of ordinary skill in the art at the time the invention to use the method of emulation of floating point operations of Erskine et al.'s in combination with the floating-point computation method of Markstein et al.

One of ordinary skill in the art at the time the invention would have been motivated to make the combination because both inventions utilizing the PA-RISC

computer architecture, which provides highly regular instruction formats (Erskine, col. 18, lines 21-27) and (Markstein, page 3, paragraph [0035]). Erskine discloses various changes could be applied to his invention (col. 14, lines 42-46). Markstein et al. also discloses his invention can be used in a wide variety of processor architectures (page 3, paragraph [0035]). Therefore use of a specific processor (e.g. Itanium™ processor) by Markstein is merely a modification to Erskine's PA-RISC computer architecture.

Claims 15 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erskine et al. (U.S. Patent No. 6,564,162) in view of Breslau et al. (U.S. Patent No. 6,101,501) and in further view of Van Dyke et al. (U.S. Patent No. 7,047,394) and Markstein et al. (U.S. PGPUB 20040158600).

As per claims 15 and 20, Erskine in view of in view of Breslau and Van Dyke fails to explicitly disclose an Itanium™ processor.

Markstein teaches:

processor represents an Itanium™ processor, said rendering said floating point hardware available including clearing at least one of a DFH and a DFL bit in a processor status register of said processor (page 3, paragraph [0035] and [0036]).

It would have been obvious to one of ordinary skill in the art at the time the invention to use the method of emulation of floating point operations of Erskine et al.'s in combination with the floating-point computation method of Markstein et al.

One of ordinary skill in the art at the time the invention would have been motivated to make the combination because both inventions utilizing the PA-RISC computer architecture, which provides highly regular instruction formats (Erskine, col. 18, lines 21-27) and (Markstein, page 3, paragraph [0035]). Erskine discloses various changes could be applied to his invention (col. 14, lines 42-46). Markstein et al. also discloses his invention can be used in a wide variety of processor architectures (page 3, paragraph [0035]). Therefore use of a specific processor (e.g. Itanium™ processor) by Markstein is merely a modification to Erskine's PA-RISC computer architecture.

Response to Arguments

Applicant's arguments with respect to claims 1-27 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Elmira Mehrmanesh whose telephone number is (571) 272-5531. The examiner can normally be reached on 8-5 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Robert W. Beausoliel, Jr./
Supervisory Patent Examiner, Art Unit 2113